Deletion and Sorting

In addition to insertion, we will also cover array deletion and sorting. Searching for a particular key in an array will also be discussed.

Overall, arrays are an important data structure in computer science, and understanding how to perform operations on arrays is essential for any programmer. By following the code examples provided in this blog post, you should have a better understanding of how to traverse an array, insert and delete data, and sort an array.

## 1.3 Array Operations | Deletion from Array | Explanation with Code | Data Structure

In this video, we will discuss the deletion operation in arrays. We will use an example array of size 5 to explain the process of deleting data from a specific position, beginning, and end of the array. We will also write the code and analyze the time complexity of the operation.

First, we initialize an array of size 50 but ask the use for the number of elements they want to insert. We then populate the array with the user's input and ask them which position they want to delete data from. For example, if they choose obsidion 2 (index 1) we cannot leave that space blank. Instead, we shift the elements to fill the empty space and decrease the size of the array by 1.

We start a loop to shift the elements to the left, starting from the index before the position chosen by the user until the second last index of the array. We shift the element at index i+1 to index i and continue until the end of the loop, overwriting the deleted element. If we want to print the deleted data, we store it in a separate variable before shifting the elements.

We provide the code for the deletion operation, where we ask the user for the position to delete, check if it's a valid position, and then start the loop to shift the elements.

```
int size = 5; int arr[size]; int pos, i; int
item; // ask user for position to delete
printf("Enter position to delete: "); scanf("%d",
```